Windows Programming
Visual C++ MFC Programming

Lecture 07

김예진
Dept. of Game Software

Notices

- 03/07: 502 → 501 등록 이동
- 03/21: HW 1 (Due: 03/28)
- 04/09: HW 2 (Due: 04/16)
- 04/25: Midterm (실습, ~75 min, 강의록 1~7)

Plan

- Image와 Timer
- MFC Timer

MFC와 Timer

주기적인 작업을 지정하기

Image와 Timer

- Still Image
  - One image

Still Life [Cézanne, Paul]

- Animation
  - Lots of images: Image sequence

Image와 Timer

- 30 → 60 → 120 frames per seconds (fps)

Timer

- SetTimer() 함수
  - 매 설정된 시간마다 WM_TIMER 메시지 발생

```
void SetTimer(int id, int time, void *fp);
```

  - id: timer의 id, 예) 0, 1, 2, …
    - timer가 여러 개 있을 경우 구분하기 위해 사용
  - time: 알람을 울릴 주기 (millisecond), 예) 1000 = 1초
  - 사용 예

```
SetTimer(0, 100, NULL);
```

  - 반드시 윈도우가 만들어 진 후 설정 해야함
    - 주로 WM_CREATE의 handler인 OnCreate 내부에 설정

Timer

- WM_TIMER handler

```
afx_msg void OnTimer(int nIDEvent)
```

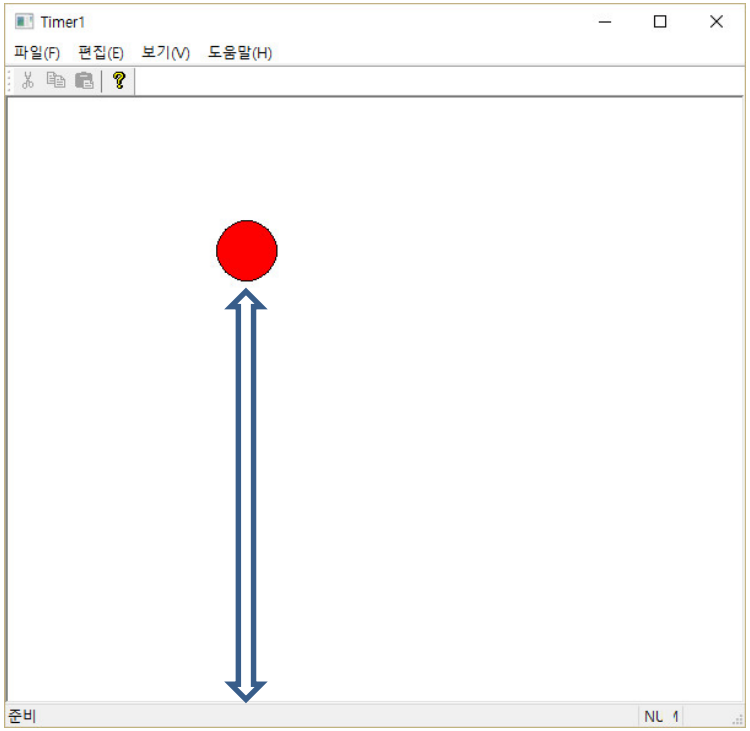
  - nIDEvent: 현재 WM_TIMER를 발생시킨 타이머의 ID
  - 사용 예

```
void CChildView::OnTimer(int nIDEvent)
{
	if (nIDEvent == 0)
	{
		// 주기 마다 해 줘야 할 일
	}
}
```

Timer

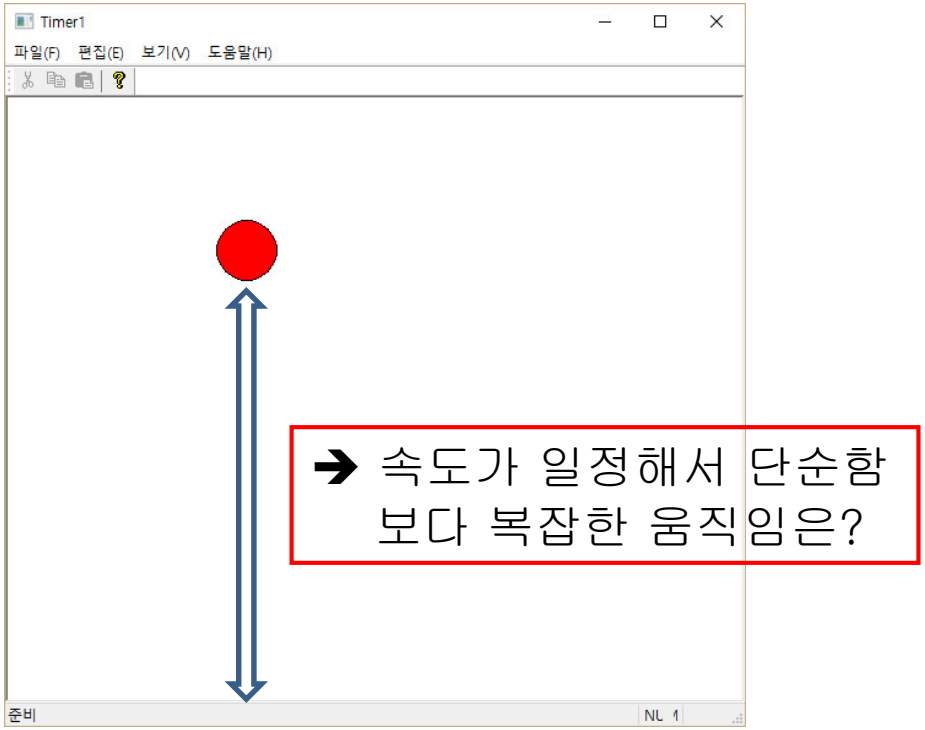
- 사용 예: 공이 상하로 움직이는 program 만들기

```
// 위치 저장 변수 선언
    CPoint m_pt;

// 위치에 원 그리기
    OnPaint()
    dc.Ellipse(m_pt.x, ...);

// WM_CREATE handler 추가
    OnCreate()

// OnCreate() 함수에 Timer 세팅
    SetTimer(0, 30, NULL);

// WM_TIMER 핸들러 추가
    OnTimer()
    if (nIDEvent == 0) {
            .....
    }
```



Timer

- 공의 움직임
  - 속도가 일정해서 공의 움직임이 단순함 → 운동학(dynamics) 추가
    - Dynamics: 도형의 위치를 의미하는 변수의 값을 시간에 따라 변화 시켜 주는 것
  - 도형의 값(values/properties)?
    - 색, 모양, 위치, etc…



Timer

- Dynamics: 물체의 운동을 기술하기 위해 필요한 값들
  - 위치(position): 보통 p 로 표현
    예) p(t) : t초 때 위치
  - 속도(velocity): 위치의 시간에 따른 변화 (dp/dt)
    예) v(t)= p(t+1) – p(t)
  - 가속도(acceleration): 속도의 시간에 따른 변화 (dv/dt)
    예) a(t) = v(t+1) – v(t)

➔ 시간에 따른 위치가 주어지면 속도 및 가속도를 구할 수 있음.
그 반대는?

Timer

- Dynamics: 물체의 운동을 기술하기 위해 필요한 값들
  - 가속도(acceleration): 보통 a 로 표현
    예) a(t): t초 때 가속도
  - 속도(velocity): t초 후 속도차이는 가속도 만큼
    예) v(t+1) = v(t) + a(t) * 1초
  - 위치(position): t초 후 위치 차이는 속도 만큼
    예) p(t+1) = p(t) + v(t) * 1초

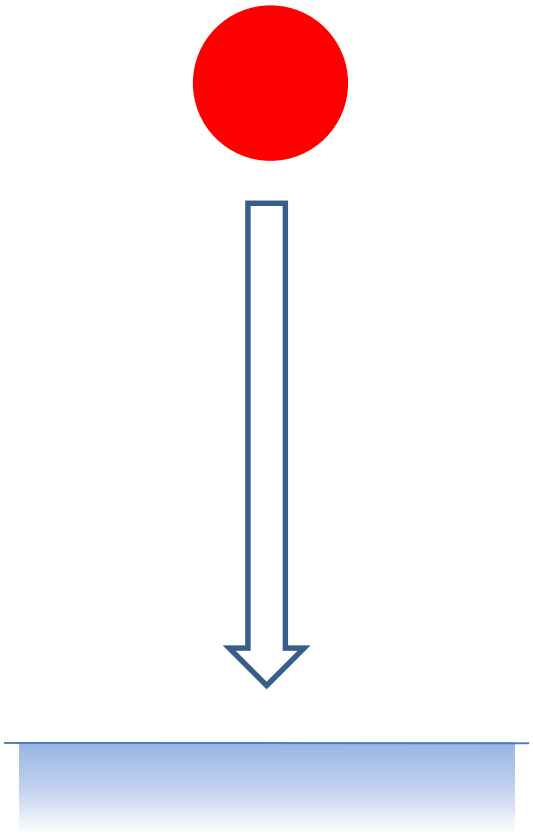
- Newton의 운동 방정식(equation of motion): **f = ma**
  - 힘(force)이 주어지면 가속도를 계산할 수 있음
  - 가속도가 주어지면 Δt 초 후의 속도가 계산 가능
  - 속도가 주어지면 Δt 초 후의 위치가 계산 가능
    예) 자유낙하: 힘 = 중력가속도 g ($-9.8 m/sec^2$) * 무게
    예) 스프링: 힘 = 기준 위치와의 차이 f = kx (k: 스프링 상수)

Timer

- 운동학을 programming
  1. 위치, 속도, 가속도를 저장할 변수를 만듬
     (m_p, m_v, m_a)
  2. 운동학(dynamics): 정해진 시간마다 다음의 일을 반복
     A. 주어진 상태에서의 힘 계산: 예) 중력 or 스프링 힘
     B. 가속도 값 갱신: a = f / m
     C. 속도 값 갱신: v = v + a * dt
     D. 위치 값 갱신: p = p + v * dt
     E. 변경된 위치에 그림 그리기

Timer

- 사용 예: 공 튕기기



아래와 같은 순서대로 각자 coding 해 보자

1. 정해진 위치에서 공이 자유 낙하
   → a = g

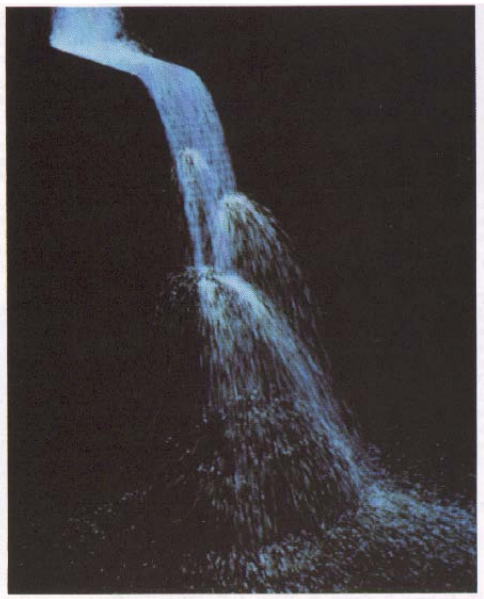
2. 마우스로 클릭하면 공의 위치를 다시 세팅

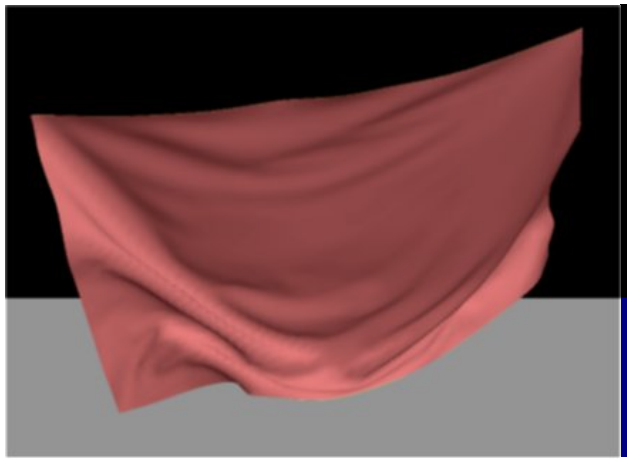
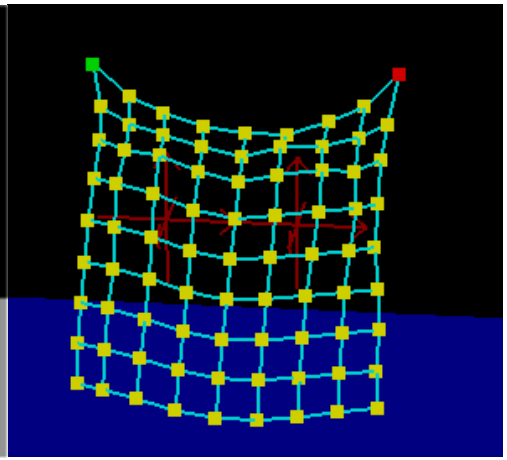
3. 밑에 벽이 있어 공이 다시 튀겨 올라감
   벽에 닿는 순간 다음과 같이 값을 변경
   → p(t + dt) = 벽과 닿은 위치
   → v(t + dt) = - e * v(t) (e: 반발계수, 보통 0.8)

4. 마우스로 공을 클릭하면 그 순간만 가속도 증가 (= 드리볼)
   → a(t) = g + f (f: 임의의 값)
   → 이 후 다시 a = g로 회귀 해야 함

Timer

- 운동학을 이용한 다양한 예제들

Q & A